



**TECNOLÓGICO DE ESTUDIOS SUPERIORES DE
ECATEPEC**

**DIVISIÓN DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

BASE DE DATOS PARA DISPOSITIVOS MÓVILES

MOSTRAR DATOS BD Y CRUD FRONTEND

PROYECTO: COMERCIALIZACIÓN

MÓDULO SELECCIONADO: CRUD DE PRODUCTOS

GRUPO: 5851

PRESENTA:

LÓPEZ ALCÁNTARA ALAN JESÚS

PROFESORA:

CORTES BARRERA GRISELDA

JULIO 11, 2022

CRUD FRONTEND DEL MÓDULO

ESPECIFICACIONES:

Proyecto: Comercialización

Integrante: López Alcántara Alan Jesús

Módulo: CRUD de productos.

Repositorio Backend: https://github.com/alan11-wq/Backend_Comercializacion

NOTA: Trabajo el proyecto individualmente y por indicaciones de la profesora seleccioné un módulo independiente a trabajar en lugar de trabajar todo el proyecto en completo.

INSTRUCCIONES:

De acuerdo con el proyecto asignado, se pide subir manual técnico donde describa los detalles para mostrar los datos de la tabla correspondiente desde la interfaz gráfica (Frontend).

Y desde esta plataforma poder realizar, altas, bajas, consultas y modificaciones del producto desde el Frontend y que pueda visualizarse en la base de datos posteriormente.

DESARROLLO:

MÓDULO PRODUCTOS

Para el desarrollo de las peticiones GET, DELETE, POST Y PUT pero en el frontend es necesario la implementación de HTTPClient, un módulo que permite la implementación de datos por el lado del cliente y que se alojan en el servidor y en la base de datos posteriormente.

VISUALIZACIÓN INICIAL DE REGISTROS EN LA BASE DE DATOS (TABLA PRODUCTOS)

```
215 • INSERT INTO Prueba (id_prueba, nombre_prueba, descripcion_prueba)
216 VALUES
217 ('701', 'Alan', 'Lopez'),
218 ('702', 'Jose', 'Martinez')
219
220 SELECT * FROM Prueba
```

id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
801	Televisor	Televisor de 70 pulgadas	\$1000	10	705
802	Audifonos Sony	Audifonos alambricos Sony	\$300	20	701
803	Play Station 5	Consola de Play Station 500gb	\$1800	8	701
804	Telefono inalambrico	Telefono inalambrico	\$400	20	704
805	Xbox one 1 TB	Consola de videojuegos xbox one	\$3,499	3	701
808	Pulsera	Pulsera electronica	200	50	702
* NULL	NULL	NULL	NULL	NULL	NULL

SERVICIO GET

Este servicio nos va a permitir la consulta de un producto, con la creación de un método de ejecución del servicio y su invocación desde un botón en la interfaz podemos consultar un nuevo producto por medio de su ID.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ConsultaServicioService {

  URL='http://localhost:5000/productos/consulta/808';

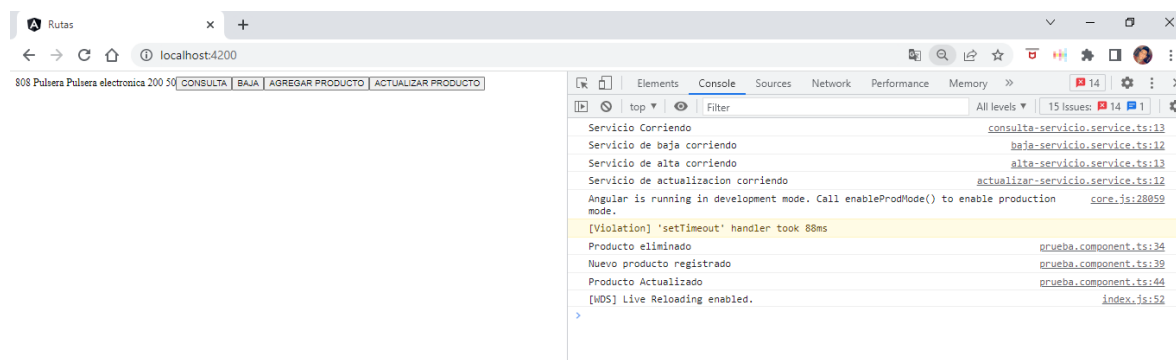
  constructor(private http: HttpClient) {
    console.log("Servicio Corriendo")
  }
  consultaBD():Observable<DatosConsulta[]>{
    return this.http.get<DatosConsulta[]>(this.URL)
  }
}

export interface DatosConsulta
{
id_producto:String,
nombre_producto: String,
descripcion_producto: String,
precio_producto: String,
stock_producto: String,
id_categoria: String
}
```

Creación del método en el archivo .ts del componente del frontend

```
EjecutaConsulta() {
  this.consultaServicio.consultaBD().subscribe(data=> this.lista = data);
}
```

Visualización en el navegador



SERVICIO DELETE

Este servicio nos permitirá eliminar un producto de la base de datos de acuerdo al ID seleccionado, para ello diseñamos un método de ejecución del servicio para que por medio de un botón en la interfaz nos permita eliminar el producto y que podamos corroborarlo desde la base de datos.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class BajaServicioService {
  URL='http://localhost:5000/productos/baja/808';

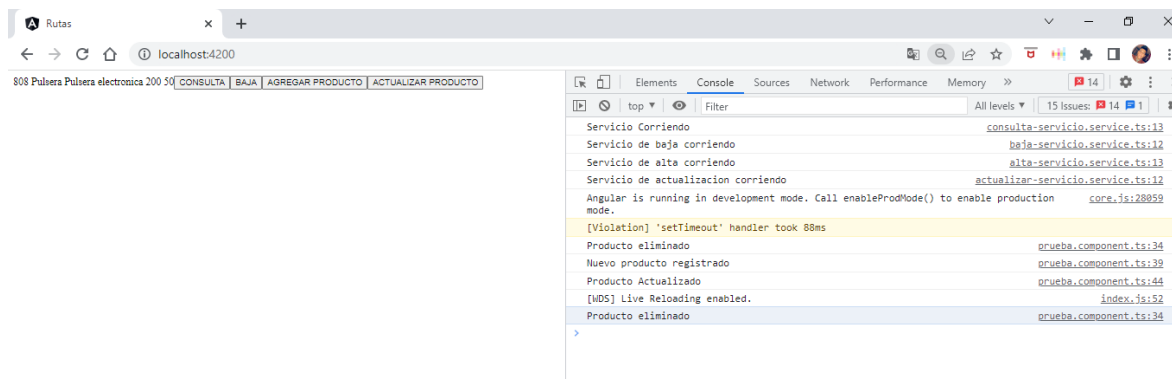
  constructor(private http:HttpClient) {
    console.log("Servicio de baja corriendo")
  }
  bajaBD():Observable<any>{
    return this.http.delete(this.URL)
  }
}

export interface DatosBaja
{
  id_producto:String,
  nombre_producto: String,
  descripcion_producto: String,
  precio_producto: String,
  stock_producto: Number,
  id_categoria: String
}
```

Creación del método en el archivo .ts del componente del frontend

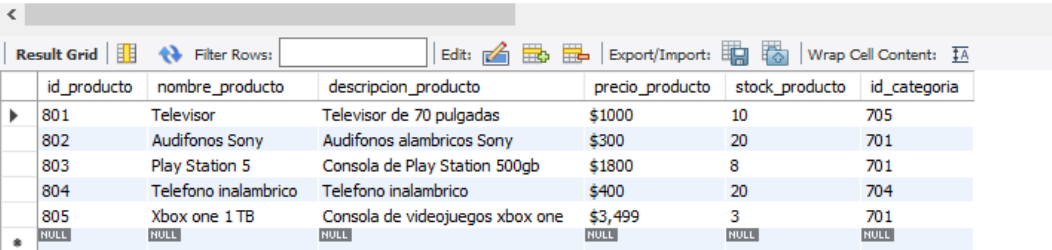
```
EjecutaBaja() {
  this.bajaServicio.bajaBD().subscribe(res=>{ console.log(res) });
}
```

Visualización en el navegador



Comprobación de la eliminación del producto en la base de datos, en este caso eliminamos el producto con el ID=808.

```
192
193 SELECT * FROM Usuarios
194 SELECT * FROM Tienda
195 SELECT * FROM Producto
196 SELECT * FROM Roles
197 SELECT * FROM Datos_Usuario
198 SELECT * FROM Direccion_Usuario
199 SELECT * FROM Compra
```



	id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
▶	801	Televisor	Televisor de 70 pulgadas	\$1000	10	705
	802	Audifonos Sony	Audifonos alambricos Sony	\$300	20	701
	803	Play Station 5	Consola de Play Station 500gb	\$1800	8	701
	804	Telefono inalambrico	Telefono inalambrico	\$400	20	704
	805	Xbox one 1 TB	Consola de videojuegos xbox one	\$3,499	3	701
*	NULL	NULL	NULL	NULL	NULL	NULL

SERVICIO POST

Este servicio nos permite la inserción de un nuevo producto desde la interfaz y que se aloja en la base de datos. Para esto se realiza la creación de un método de ejecución del servicio y que por medio de un botón nos permite generar un nuevo registro de un producto.

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AltaServicioService {
  URL='http://localhost:5000/productos/alta'

  constructor(private http:HttpClient) {
    console.log("Servicio de alta corriendo");
  }

  altaBD(alta:DatosAlta):Observable<DatosAlta>{
    return this.http.post<DatosAlta>(this.URL, alta);
  }
}

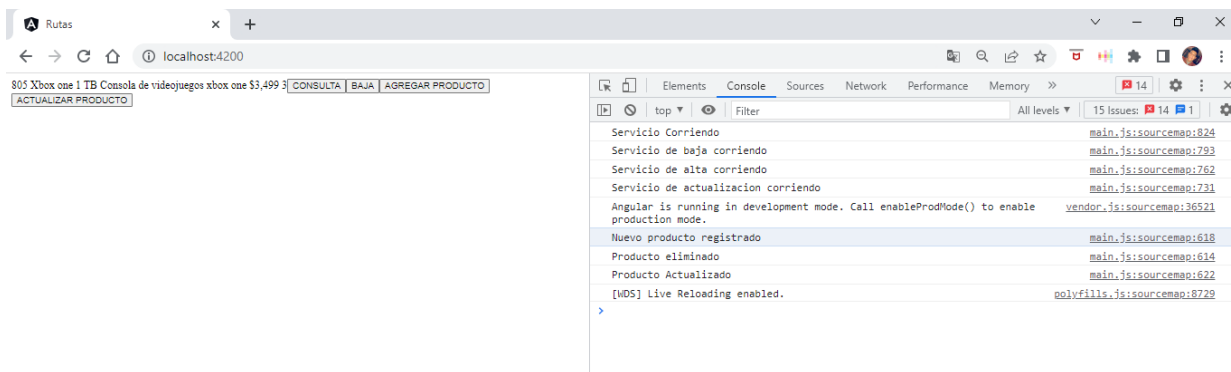
export interface DatosAlta{
  id_producto:String,
  nombre_producto: String,
  descripcion_producto: String,
  precio_producto: String,
  stock_producto?: String,
}
```

```
id_categoria: String
}
```

Creación del método en el archivo .ts del componente del Frontend.

```
EjecutaAlta() {
  const altaProducto = {id_producto:'809', nombre_producto:'Licuadora
Koblenz', descripcion_producto:'Licuadora de 6
velocidades', precio_producto:'$2,999', stock_producto:'10', id_categoria:'7
02'}
  this.altaServicio.altaBD(altaProducto).subscribe(data=>
console.log(data));
}
```

Visualización en el navegador.



Comprobación de la inserción del nuevo registro en la base de datos, en este caso se insertó el producto con el ID=809.

```

193 SELECT * FROM Usuarios
194 SELECT * FROM Tienda
195 SELECT * FROM Producto
196 SELECT * FROM Roles
197 SELECT * FROM Datos_Usuario
198 SELECT * FROM Direccion_Usuario
199 SELECT * FROM Compra

```

	id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
	801	Televisor	Televisor de 70 pulgadas	\$1000	10	705
	802	Audifonos Sony	Audifonos alambricos Sony	\$300	20	701
	803	Play Station 5	Consola de Play Station 500gb	\$1800	8	701
	804	Telefono inalambrico	Telefono inalambrico	\$400	20	704
	805	Xbox one 1 TB	Consola de videojuegos xbox one	\$3,499	3	701
▶	809	Licuadora Koblenz	Licuadora de 6 velocidades	\$2,999	10	702
	810	Monitor PC	Monitor de PC de 40 pulgadas	\$5,999	10	702
*	NULL	NULL	NULL	NULL	NULL	NULL

SERVICIO PUT

Este servicio nos permite la actualización de alguno de los campos de cualquier producto registrado en la base de datos de acuerdo a su ID, para ello realizamos la creación de un método de ejecución del servicio que por medio de un botón en la interfaz permita enviar las actualizaciones y estas se corroboren en la base de datos.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ActualizarServicioService {
  URL = 'http://localhost:5000/productos/actualizar'

  constructor(private http: HttpClient) {
    console.log("Servicio de actualizacion corriendo")
  }

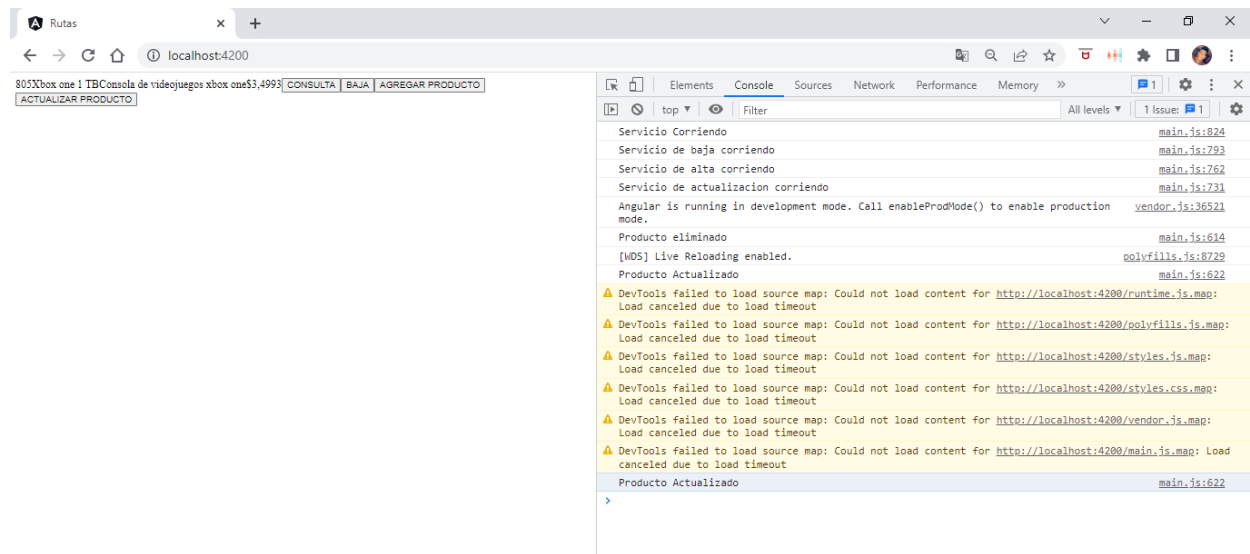
  actualizaBD(update:DatosActualizar): Observable<DatosActualizar> {
    return this.http.put<DatosActualizar>(this.URL,update);
  }
}

export interface DatosActualizar
{
  id_producto:String,
  nombre_producto: String,
  descripcion_producto: String,
  precio_producto: String,
  stock_producto: String,
  id_categoria: String
}
```

Creación del método en el archivo .ts del componente del Frontend. Este archivo contiene los datos a modificar.

```
EjecutaActualizar(){
  const updateProducto=
{id_producto:'810',nombre_producto:'Pulsera',descripcion_producto:'Pulsera
electronica',precio_producto:'200',stock_producto:'50',id_categoria:'702'
}
  this.actualizaServicio.actualizaBD(updateProducto).subscribe(data
=>{ console.log(data) });
}
}
```

Visualización en el navegador.



Comprobación de la actualización del producto con el ID=810 en la base de datos con los nuevos valores asignados en el método.

```
174
193 SELECT * FROM Usuarios
194 SELECT * FROM Tienda
195 SELECT * FROM Producto
196 SELECT * FROM Roles
197 SELECT * FROM Datos_Usuario
198 SELECT * FROM Direccion_Usuario
199 SELECT * FROM Compra
```

	id_producto	nombre_producto	descripcion_producto	precio_producto	stock_producto	id_categoria
▶	801	Televisor	Televisor de 70 pulgadas	\$1000	10	705
	802	Audifonos Sony	Audifonos alambricos Sony	\$300	20	701
	803	Play Station 5	Consola de Play Station 500gb	\$1800	8	701
	804	Telefono inalambrico	Telefono inalambrico	\$400	20	704
	805	Xbox one 1 TB	Consola de videojuegos xbox one	\$3,499	3	701
	809	Licudora Koblenz	Licudora de 6 velocidades	\$2,999	10	702
	810	Pulsera	Pulsera electronica	200	50	702
*	NULL	NULL	NULL	NULL	NULL	NULL

Código HTML del componente del Frontend.

```
<div *ngFor="let i of lista">
  <label>{{i.id_producto}}</label>
  <label>{{i.nombre_producto}}</label>
  <label>{{i.descripcion_producto}}</label>
  <label>{{i.precio_producto}}</label>
  <label>{{i.stock_producto}}</label>
  <Button (click)="EjecutaConsulta()">CONSULTA</Button>
  <Button (click)="EjecutaBaja()">BAJA</Button>
  <Button (click)="EjecutaAlta()">AGREGAR PRODUCTO</Button>
  <Button (click)="EjecutaActualizar()">ACTUALIZAR PRODUCTO</Button>
</div>
```


Archivo del componente .ts

```
import { Component, OnInit } from '@angular/core';
import { ConsultaServicioService, DatosConsulta } from
"../servicios/consulta-servicio.service";
import { BajaServicioService, DatosBaja } from "../servicios/baja-
servicio.service";
import { AltaServicioService, DatosAlta } from "../servicios/alta-
servicio.service";
import { ActualizarServicioService, DatosActualizar } from
"../servicios/actualizar-servicio.service";

@Component({
  selector: 'app-prueba',
  templateUrl: './prueba.component.html',
  styleUrls: ['./prueba.component.css']
})
export class PruebaComponent implements OnInit {

  lista: DatosConsulta[] = [];
  listabaja: DatosBaja[] = [];
  listaalta: DatosAlta[] = [];
  listaactualizar: DatosActualizar[] = [];

  constructor(private consultaServicio: ConsultaServicioService, private
bajaServicio: BajaServicioService, private
altaServicio: AltaServicioService, private
actualizaServicio: ActualizarServicioService) { }

  ngOnInit(): void {
    this.EjecutaConsulta()
    this.EjecutaBaja()
    this.EjecutaAlta()
    this.EjecutaActualizar()
  }

  EjecutaConsulta() {
    this.consultaServicio.consultaBD().subscribe(data => this.lista =
data);
  }

  EjecutaBaja() {
    this.bajaServicio.bajaBD().subscribe(res => { console.log(res) });
  }

  EjecutaAlta() {
    const altaProducto = { id_producto: '810', nombre_producto: 'Monitor PC
', descripcion_producto: 'Monitor de PC de 40
pulgadas', precio_producto: '$5,999', stock_producto: '10', id_categoria: '702'
}
    this.altaServicio.altaBD(altaProducto).subscribe(data =>
console.log(data));
  }

  EjecutaActualizar() {
    const updateProducto =
```

```
{id_producto:'810',nombre_producto:'Pulsera',descripcion_producto:'Pulsera
electronica',precio_producto:'200',stock_producto:'50',id_categoria:'702'
}
  this.actualizaServicio.actualizaBD(updateProducto).subscribe(data
=>{ console.log(data) });
}
}
```